



GLI®

World Headquarters

600 Airport Road
Lakewood, NJ 08701

Phone (732) 942-3999
Fax (732) 942-0043
www.gaminglabs.com

November 4, 2021

Ms. Susan Christie, Sr. Manager North American Product Compliance
International Game Technology (IGT)
6355 S. Buffalo Drive
Las Vegas, Nevada 89113
U.S.A.

RE: Random Number Generator Report

File Nr: RN-337-IGT-21-01

IGT Ref: PSR-20477

Dear Ms. Christie,

Enclosed, please find a detailed explanation of the Random Number Generator (RNG) testing results of the IGT 'Remote Gaming System (RGS) RNG', evaluated against the applicable RNG-specific requirements listed herein.

This test report may not be reproduced, other than in full, except with the prior written permission of Gaming Laboratories International, LLC.

Please visit Gaminglabs.com to view the applicable Terms and Conditions and GLI Product Certification Scheme.

Registration Number of Accreditation applicable to this Report:	A2LA 2428.05
---	--------------

If you should have any questions regarding this Random Number Generator Report, please feel free to contact our office.

Sincerely,

GAMING LABORATORIES INTERNATIONAL, LLC

Christine M. Gallo
Senior Vice President, Quality and Technical Compliance

nbm

Worldwide Locations

World Headquarters
Lakewood, New Jersey

U.S. Regional Offices
Colorado
Nevada

International Offices
GLI Africa
GLI Asia
GLI Australia Pty Ltd
GLI Austria GmbH
GLI Europe BV
GLI Italy
GLI South America



RANDOMNESS REPORT FOR THE IGT REMOTE GAMING SYSTEM (RGS) RNG

The intent of this report is to indicate that **Gaming Laboratories International, LLC (GLI)** has completed its evaluation of the IGT RGS random number generator (RNG), version 2.0.3, provided by IGT.

SECTION I - SCOPE OF TESTING

GLI was provided the required materials in order to conduct a random number generator analysis on the IGT RGS RNG. The scope of this analysis was limited to software verification, source code review, and data analysis. The RNG was tested for its ability to randomly produce outcomes for the IGT RGS games.

The IGT RGS RNG was evaluated against the RNG-specific requirements of the following technical standards for the following technical standards:

- Gibraltar: Remote Technical and Operating Standards for the Gibraltar Gambling Industry - v1.1.0
- Gibraltar: Gambling Commissioner's Guidelines - v.1.0.2012
- Gibraltar: Gambling Act 2005 - Act. No. 2005-72 Commencement (LN. 2006/114) except for s. 55(b) 26.10.2006 Assent 22.12.2005.
- UK Remote: The Gambling Commission's Remote Gambling and Software Technical Standards, February 2021
- UK Remote: The Gambling Commission's Testing Strategy for Compliance with Remote Gambling and Software Technical Standards, February 2021



SECTION II - SOFTWARE VERIFICATION

Verify+ by Kobetron™ signatures for the IGT RGS RNG are as follows:

File	Type	Signature
IRNGService-2.1.0-SNAPSHOT.jar ^①	Kobe4	P783
	MD5	932C3861A2F35AA74C3DFA4B0CCF1966
	SHA-1	180FE63733F234AE4AC89D88D8367152E1AE3C7A
ng-common-2.0.3.jar ^①	Kobe4	45P6
	MD5	CDD32B07314FD83F6538FC1EA3F9CC80
	SHA-1	F54619873DCA4CFCE01776CB14FB66E6557B13EC
ng-core-2.0.3.jar ^②	Kobe4	23P4
	MD5	777B9679AFFA0E7015F8C31FF9B9C8C1
	SHA-1	FE4DBC5E41E2B5C213F4B936F20689E32593B521
ng-domain-2.0.3.jar ^①	Kobe4	UP75
	MD5	ABBA4871E4EB78A0FE18F7414127BF1E
	SHA-1	BF0F05A18705A3E97B38F28F9774C8C577C3B308
ng-monitoring-2.0.3.jar ^①	Kobe4	89A1
	MD5	438D981B4D3E76B5F50DC6452BA6F0E1
	SHA-1	07392FDD34B72C2A34612F53251ED52E06BDA477
RNGProxy-2.1.0-SNAPSHOT.jar ^①	Kobe4	2A71
	MD5	F978148D9DCE4365A97780A2EB4CB9A6
	SHA-1	FC7A8DEEFC01EDD73B408210DCDDFD51B6A87B30

Table 1. Digital Signatures For .jar Files

① The RNG 2.0.3 (and 2.1.0) critical files contain methods that support the ng-core-2.0.3.jar critical file. The signatures listed herein describe the files provided for evaluation. Although the content of the .jar files may remain the same, the signatures may vary after each build due to the build process. Therefore, the signatures herein will possibly not match the eventual production environment.

② The RNG 2.0.3 critical file contains the core RNG methods, which will be called by the games in order to produce random outcomes. The signatures listed herein describe the files provided for evaluation. Although the content of the .jar files may remain the same, the signatures may vary after each build due to the build process. Therefore, the signatures herein will possibly not match the eventual production environment.



Since the signatures for the ng-core-2.0.3.jar file may vary, the .class files were included in a separate chart as the signatures will not change for these if they are not modified. These can then be used to verify that the correct .class files are within the ng-core-2.0.3.jar file. The signatures for the critical .class files are as follows.

File	Type	Signature
AbstractKeyGenerator.class	Kobe4	C92C
	MD5	6BC48E66E986EF141E67BE54D5CAF6C9
	SHA-1	88C3B348F6C249CB151731F99ED68D1965F127C5
AbstractMonitor.class	Kobe4	63P0
	MD5	BDF0C65D1AD84980C77A48518133544B
	SHA-1	597CBE2855D8F78661E6CA964FBC1D160512C051
AbstractNG.class	Kobe4	CUF3
	MD5	51E88582E3C02B2F481B1869EE407A1F
	SHA-1	D332D2CB82A84CD94AF19B2F981E409B26570742
AbstractPostProcessor.class	Kobe4	F065
	MD5	EC79F16811C040CB6627C01B3BF81DA7
	SHA-1	DFCA41D7F27930DCAA9B026F9E0FEE82389CF35B
AbstractPreProcessor.class	Kobe4	7020
	MD5	D21E83DE4916F47E8718E53F95462E14
	SHA-1	A885223D0AACF5741DBB156C42A617AE1E3DF1DC
AbstractPullProcessor.class	Kobe4	16PF
	MD5	C0CA5ED7A75BB9D0F48D43365DD539A0
	SHA-1	AE186CCD60417AB2082BD5F1A234A41F3B6A06E4
AbstractRequestReader.class	Kobe4	2CF9
	MD5	01BC7D7712AC56CE6B4F27F4B44D4633
	SHA-1	8D630235A8CBDC89C3E61F74FCD3561C3A500C59
AuditorRingBuffer.class	Kobe4	2353
	MD5	D247164D67030B1FBA1221E4EF633AA1
	SHA-1	D13197AF0649C6265621EE0D3097E710D9359480
AutoSeed.class	Kobe4	U9H5
	MD5	DFA4C1C01E49A878AB4C50CDE7AE97C6
	SHA-1	CEFAB369AF039138E6B9B646B5AEFC232C4C2214



File	Type	Signature
BigNum.class	Kobe4	6C44
	MD5	32B122E7907DE360CA6BCBB144C51C6C
	SHA-1	CBD406FBD7C6EAB19E682EB5B0FEDBEBA05828D6
BitCalculationProcessor.class	Kobe4	3759
	MD5	3363A0B6A955BEF03DB8C7A0A02CCCCF
	SHA-1	0FA232A88A3AA9165B27EC67019AB0E09520D03C
BucketEventHandler.class	Kobe4	9AF4
	MD5	8E1BB0BA3AD1602F85F4C8A043BB4F16
	SHA-1	6215BDD1DC25806EB01D5E94B3692B403AE9047F
Buckets.class	Kobe4	U35C
	MD5	4A629881A076B6693CDB2FEB771F3A69
	SHA-1	F57E919EEEEF97B3F60FEEF340A78430386B72C1C
ChiSquareCalculator.class	Kobe4	616C
	MD5	31BBDE4CC0F15554BC20AFC54D56A8F6
	SHA-1	7662CEF375E086D23F0D00AE1C1DF829938F2D1B
DuplicateAuditorEventHandler.class	Kobe4	FC15
	MD5	AC8AD2C1ABBD4F987675E2D86D874011
	SHA-1	5BE59AFA583586C518A150EEFF541CD1EA590ACC
DuplicatesAuditor.class	Kobe4	0C6P
	MD5	E30B9252C856FD90A63BC85CFE1A2958
	SHA-1	9FA997BE8AF63B023B0CDC2D13C900601BCE1A2E
FullCycleRNG.class	Kobe4	6C1C
	MD5	BC5E6F42277D2BFE80188C116445C773
	SHA-1	188DB11043E7EB3744EAE22B882D7442762D89FE
IAuditor.class	Kobe4	34P3
	MD5	9364D3573F1460EA885B6E3E384D0D66
	SHA-1	437E1CE694CDED6A96675E525F3BB3F1A57897F2
IFullCycleSeed.class	Kobe4	561U
	MD5	C5D62D4221D3C70C9CF5C4A16A728706
	SHA-1	E88EED085A3C1D839234011F4491EB24960F73B9



File	Type	Signature
IKeyGenerator.class	Kobe4	H79C
	MD5	544B8616AFAD8DAA6CD61459C756C000
	SHA-1	35CAC08B932595B071EF3C86FC63484C1703E245
IMarsagliaSeed.class	Kobe4	8589
	MD5	55C755BFC121774F20E73283FB53D686
	SHA-1	798BA331A3BBDDD8CD7C471DFFCD8269EBC4FD20
IMonitor.class	Kobe4	PP44
	MD5	7E90789A4AF3A146E06D81F24F18FE23
	SHA-1	13A7DF4EBD009486627CE3D1EFEFC4B6476A4A99
ING.class	Kobe4	U089
	MD5	262A786EF29B82B7F3D89592B41C30D6
	SHA-1	9041CF0FF78D6B0D99796A38375D0A8563D8918E
IPreProcessor.class	Kobe4	9F9U
	MD5	296DF601AEA9BDD5BBD33DE040DB8FC3
	SHA-1	5895C675F6DB73C356B61E169D14D7E8775B3252
IPullProcessor.class	Kobe4	HFF6
	MD5	ECCADC78FD8E1B589733A02A31FE6CD1
	SHA-1	317477F4064D8132C96458B6961227ABEEF56E8E
ISeed.class	Kobe4	H212
	MD5	6829452C92ABF5D2F16E92219A7EE5E8
	SHA-1	EA3FD206828A4C3E296AE0430B843DA9AF911EEE
LongsEvent.class	Kobe4	4FHH
	MD5	840FF5B2AA4483DD98C2C4D03C72E4D5
	SHA-1	75DFE9AC0DDD0450ADE090BD7133F31861EDC676
MarsagliaAutoSeed.class	Kobe4	U162
	MD5	EB4DDA4890ED6AAFEEA34D881B24254C
	SHA-1	98B1EE654418A477E4D797261D79C04FC4E2C1EB
MarsagliaRNG.class	Kobe4	H169
	MD5	831083C468DF88BF2B96D3EF4A84F1EA
	SHA-1	31F3ECD9AC6EAB844CED3B2049496CBF2E42C00A



File	Type	Signature
NGCycleDaemon.class	Kobe4	396C
	MD5	FACA66693EF5C0C09E6E58C83026B5DF
	SHA-1	2E62A0410BB1DD59D48729AB17DF9119AB97FCB8
NGEncryption.class	Kobe4	92HC
	MD5	0E4D5FF3A2FC0E1DFAC2D41DAD63E82F
	SHA-1	05EF0BC3C65B54FD5048B79123AE54C25D71BD42
NGLogger.class	Kobe4	0ACU
	MD5	3BA4DBC6DB9EEAFC79B28AD59E6DCC4
	SHA-1	573CDA AFC3885623C8F12ABF383951425CFBA795
NGManager.class	Kobe4	0P01
	MD5	097D94FC5FBA0DB5712D001206C89F0E
	SHA-1	4C61AD3CCDDF1A5C395F4C7A9FCA23318AA559D8
NGMonitor.class	Kobe4	7660
	MD5	02D51A234D1BF4C28B933E2D2D06E7EF
	SHA-1	99287EFFAE6272E7BA31E42A3D27ECFE7397AF38
NGTable.class	Kobe4	HP04
	MD5	127929FEC551B285184204F61739BCF4
	SHA-1	C30B91894EA2435E835C78688591900BB33CA6CE
PreProcessorMonitor.class	Kobe4	CCP6
	MD5	F048ACB50CEE806EB3B6AEFB6523CEB9
	SHA-1	FD00FA8F5BA574E60D27E5C17EAED27E6F6449C
Pull.class	Kobe4	9H73
	MD5	B99A1E4D51391D2DCED0629C849A9EB7
	SHA-1	E881E037DE238F1F87200A614550223A1C08888C
PullBuilder.class	Kobe4	490P
	MD5	0CF744F546B69C4E2FB5242700FD52A1
	SHA-1	9BE08FD39CC4060E9B36450D08383B5EABAD5E8F
PullContextProcessor.class	Kobe4	FC25
	MD5	03E76AF6F582B27719B1459495CEEDA8
	SHA-1	6AC26BB7033B6BB33643A690D824E0729893431F



File	Type	Signature
PullProcessor.class	Kobe4	H84A
	MD5	50FC58265AFF215CFCA8DD13D04776CE
	SHA-1	C808A9CDC82F886F3F52BA1D223449B10B359FA3
Request.class	Kobe4	56FC
	MD5	01D85D7AED7790BCE487E2B8DA46A85A
	SHA-1	E3DEBD7E5B2E75CF2C3D4F59B148AD3279E3B4E2
RequestBuilder.class	Kobe4	73H5
	MD5	0C87461C51B5A5F17B20A9B2D83B08D1
	SHA-1	A51A131F05076136493179626161252733825D4A
RequestByteArrayReader.class	Kobe4	6484
	MD5	E8ED53A79C944076C97147B412D62CFF
	SHA-1	F72266AA16859BE9DC62886DCDA46D31B20A1F95
RequestInputStreamReader.class	Kobe4	6PF1
	MD5	54DCFC0C0182CF061EE6018BBF0D1ED1
	SHA-1	4524678FC35E90A3149417E324B713564FFB5D90
RequestWriter.class	Kobe4	FP6P
	MD5	E34309AA8695B37C3319AA95558EB89A
	SHA-1	F2E1B9039E03B897AC8412E1742F4C8B42A46E7E
SecureRandomKeyGenerator.class	Kobe4	915U
	MD5	17F72C410F942D8D9166190A12FA5549
	SHA-1	FCA97807C039DF5143C8FBE10EB496ACE7A18B58
SensorKeyGenerator.class	Kobe4	09H5
	MD5	BF6DC9B2A943C597EBA9DAD5BB4E2686
	SHA-1	FF4C94CAADDAABF452FDBCE48455C9E03CCB0E01
TableProcessor.class	Kobe4	A808
	MD5	834B640FBFB9D8A30367D214E14091AE
	SHA-1	C356FE9265D182642561D5D9E3B82B5F48917F45
ValidationProcessor.class	Kobe4	P701
	MD5	D6D23C95F7398B8A4E80235FB402892A
	SHA-1	637470385B316D563BCE69D47F37EC8A339817DA



File	Type	Signature
	Kobe4	8H39
ValidPullCountProcessor.class	MD5	A4BFF29D4D99370C57D60661EA39DD54
	SHA-1	E589B797E12151E9C79B4BF40716A14502DDDD34

Table 2. Digital Signatures For .class Files

SECTION III - SOURCE CODE REVIEW

GLI received the appropriate documentation and full source code which pertains to the generation of random numbers. GLI reviewed the source code provided by tracing the path of the RNG application from the initiation of the draw to the selected output of random numbers. GLI inspected the source code, where practicable, in an attempt to find any undisclosed switches or parameters having a possible influence on randomness and fair play. GLI assessed the ability of the RNG to produce all numbers within the desired range.

SECTION IV - DATA ANALYSIS

The game configuration and parameters for the data obtained and tested are listed in Table 3. GLI performed a data format check on each data set listed in order to confirm that the game parameters were correctly represented in the data analyzed.

GLI conducted a statistical analysis of sufficient scope to test the RNG for selecting as many as 20 winners from a pool size as large as 2,147,483,647, and as many as 520 winners from a pool size of 520 as described in Table 3. To provide this level of assessment, GLI selected different test cases for statistical testing. The selection of test cases took into account broad coverage of range sizes and selections.

A set of numbers is said to be drawn *with replacement* if a number can be selected multiple times within the same draw. A set of numbers is said to be drawn *without replacement* if a number can only be selected once within the same draw.



Data Set	Range	Positions	Replacement	Draws
General Certification	Up to and including 2,147,483,647	Up to and including 20	N/A and Yes	③
General Certification	Up to and including 520	Up to and including 520	No	③
Set92	0 - 2,147,483,646	5	No	50,000,000
Set93	0 - 390,624	5	No	200,000,000
Binary	0 – 65,535	1	N/A	6,000,000

Table 2. RNG Parameters

③ Data sets of different ranges and draw sizes were collected and analyzed to cover the scope of this general certification.

All data sets used for statistical testing were collected by IGT without GLI’s supervision and made available to GLI on Microsoft’s OneDrive file sharing system. Signature verifications within the IGT and GLI environments confirmed that no data corruption occurred during the downloading of the files to GLI.

For a summary of the statistical tests applied to each data set, see *Appendix A*. For a description of the overall test methodology and a description of each test used, see *Appendix B*.

Overall, the RNG passed the battery of tests for each configuration at the 95%, 98%, and 99% confidence levels.

SECTION V - SUMMARY

Overall Evaluation of the Random Number Generator

GLI’s conclusion based upon the tests applied to the IGT RGS RNG data is that this RNG has exhibited random behavior and is suitable for the applications as described herein. If a game utilizes different RNG parameters than the ones listed in this report, the RNG should be resubmitted to test that set of parameters.



APPENDIX A: Statistical Test Summary

Data Set	Range	Positions	Replacement	Draws	Test Names												
					Runs	Serial Correlation	Interplay Correlation	Adjacency Max-Min	Adjacency High-Low	Adjacency Blocks	Coupon Collector	Duplicates	Overlaps	Permutation	Total Distribution	Total Distribution by Position	Count of Counts
General Certification	Up to and including 2,147,483,647	Up to and including 20	N/A and Yes	③	X	X	X	X	X		X	X	X		X	X	X
General Certification	Up to and including 520	Up to and including 520	No	③	X	X	X	X	X	X	X	X	X	X	X	X	
Set92	0 - 2,147,483,646	5	No	50,000,000	X	X	X		X			X		X	X	X	
Set93	0 - 390,624	5	No	200,000,000	X	X	X	X	X	X		X	X	X	X	X	X
Binary	0 – 65,535	1	N/A	6,000,000													X

Table A 1. Tests Applied

③ Data sets of different ranges and draw sizes were collected and analyzed to cover the scope of this general certification.



APPENDIX B: Test Descriptions

B.1 Definitions. The following terms apply to the below test descriptions. Randomness Device or Random Number Generator (RNG) output may be collected multiple numbers at a time. Each set of numbers is called a draw. Each individual number has a particular order within the *draw*. This is referred to as the number *position*.

B.2 Distribution Comparisons. Many of the tests compare an observed numerical distribution with an expected distribution. Unless otherwise specified, this is done by means of a statistical chi-square goodness-of-fit test. The value chi-square is computed in the standard way. If k is a possible value, o_k is the observed count of that value, and e_k is the expected count:

$$\chi^2 = \sum_k \frac{(o_k - e_k)^2}{e_k}$$

In the case where expected counts are too small for accurate use of the above formula, values are 'binned' together to ensure an appropriate minimum expected count. The resultant value for chi-square is compared against the distribution for the appropriate number of degrees of freedom. Unusually high (distribution mismatch) or unusually low (insufficient randomness) chi-square values can be causes for data failure.

B.3 Meta-testing. Evaluation of groups of p -values may include a meta-test for extremity of high or low p -values, a meta-test for frequency of high or low p -values, and a meta-test for uniformity of p -values, as appropriate.

B.4 Confidence Level. The statistical tests conducted by GLI are done at a particular *confidence level*. Common confidence levels used include 95%, 98%, and 99%, depending on jurisdictional requirements, and intended use of the RNG. High confidence level testing has low risk of mistakenly failing a good RNG, but higher risk of passing a bad RNG. Lower confidence level testing has increased power of detecting bad RNGs, while also increasing the risk of false failures of good RNGs. Specifically, the confidence level represents the probability that an ideal source of randomness would pass the testing. If an RNG passes statistical tests at a given confidence level, passage at all *higher* confidence levels is implied.

B.5 Tests. Some tests are only applicable to certain types of data. Some tests may be applied only to a portion of the data. Some tests may require that the data be parsed, binned, or otherwise transformed, as necessitated by data format.



Adjacency Blocks:

For each draw, the data is first sorted. Then the amount of contiguous blocks of numbers is counted. These statistics are then compared against the expected. For example, if a draw consists of the numbers

1, 5, 4, 2, 6, 9

the data would be sorted and separated into blocks. The resulting statistic would be 3.

Adjacency High-Low:

For each draw, the number of local extrema ('highs' and 'lows') in the data is recorded and compared with the expected distribution. These are also referred to as 'turning points'. For example, if a draw consists of the numbers

1, 3, 5, 7, 2, 9

there would be one local maximum (7) and one local minimum (2). The resulting statistic would be 2.

Adjacency Max-Min:

For each draw, the difference between the maximum and minimum values is calculated and recorded. This is compared with the expected theoretical distribution. For example, if a draw consists of the numbers

2, 3, 6, 7, 4

the resulting statistic would be 5, the difference between the maximum value (7) and the minimum value (2).

Count of Counts:

The Count of Counts test first counts the occurrences of each value in each position of the data. These counts are then tallied and compared with the expected distribution of counts for the draw size and range of values.



Coupon Collector's:

The Coupon Collector's Test is applied positionally. The data is parsed until all possible values have been observed, then the number of values checked is recorded and the count is restarted. This is compared with the expected distribution. For example, if the set of all possible values is {0, 1, 2} and the first position of each draw is

1, 0, 1, 0, 2, 0, 1, 2, ...

then all values are observed in the first position by the fifth draw. All values are then observed within the next 3 draws, so the first two statistics for the first position would be 5 and 3.

DieHard:

The DieHard Battery of Tests is a standard assessment of the randomness in raw outcomes generated from an RNG. The collection, designed by George Marsaglia, tests for a variety of patterns in the individual binary bits of RNG output. GLI uses a custom implementation to conduct DieHard testing.

Duplicates:

The Duplicates Test counts the number of times a draw is exactly duplicated in the data. In the case that a particular draw is repeated more than twice, every possible way to generate a duplicate is counted. This is compared against the theoretical distribution to verify that the number of duplicate draws falls within expected bounds. For example, consider the dataset consisting of the following draws of two numbers each.

- a) 1, 3
- b) 4, 1
- c) 1, 3
- d) 1, 3
- e) 4, 1
- f) 3, 1

The duplicate pairs are (a, c) , (a, d) , (c, d) , and (b, e) , for a total of 4 duplicates. (f) is not counted as a duplicate since the draw must match in order as well as values.



Interplay Correlation:

The Interplay Correlation Test measures statistical correlation between different positions of the same draw. For each pair of positions, statistical correlation is calculated as in the Serial Correlation Test. In the case of without replacement data, an adjustment is made to account for the expected resulting negative correlation.

Overlaps:

The Overlaps Test compares consecutive draws for overlapping values. The number of overlapping values is recorded for each pair of draws. This observed distribution of overlaps is then compared against the expected distribution. For example, if the following draws are observed consecutively,

- a) 1, 4, 5, 6
- b) 4, 1, 7, 6

the number of overlaps would be 3, representing the values 1, 4, and 6.



Permutation:

The Permutation Test is a test applicable to data that represents a reordering of numbers. Each draw can be considered as a permutation of the original ordering. Every permutation can be decomposed into disjoint cycles, which represent the possible positions a number would occupy if the same permutation is applied repeatedly. For each draw, three statistics are collected based on the cycle decomposition:

- The number of cycles.
- The size of the smallest cycle.
- The size of the largest cycle.

Each of these statistics generates a distribution of observations which are compared with their respective expected distributions. For example, if the following draw were observed as a reordering of the numbers from 1 to 6,

1, 3, 5, 4, 2, 6

the cyclic decomposition would be (1)(2 3 5)(4)(6). 1, 4, and 6 remain in their original positions, so they form their own cycles. The values 2, 3, and 5 are shuffled, so they form a single cycle together. The total number of cycles is 4, the smallest cycle has size 1, and the largest cycle has size 3.

Runs:

The Wald-Wolfowitz Runs Test is applied to each position within the draw. A center is established, typically the data median, and the number of 'runs' above and below the center are tallied. Values exactly equal to the center are discarded. This is compared to the expected distribution, which depends on the number of values above and below the center. For example, if the numbers drawn at a particular position were

2, 3, 1, 5, 4, 7, 3, 2, 3, 2, 3, 2, 6, 7, 3, 5

and the established center were the data median of 3, the data would be parsed for runs above 3 and runs below 3.

2, 3, 1, 5, 4, 7, 3, 2, 3, 2, 3, 2, 6, 7, 3, 5

This would be counted as 4 runs.



Serial Correlation:

The Serial Correlation Test measures statistical correlation between consecutive draws of the same position. For each position, the sample Pearson correlation coefficient is calculated. If X represents the first number, and Y the number that follows, then the coefficient is

$$r = \frac{cov(X, Y)}{s_X s_Y}$$

where s denotes the sample standard deviation. The coefficients are used to generate a p -value for each position.

Total Distribution:

The Total Distribution Test is a simple tally of all observed values throughout the data. This is compared with the expected distribution. Typically, the expected distribution is a uniform distribution. In the case of unequal weighting of values, an appropriate discrete distribution is used.

Total Distribution by Position:

The Total Distribution by Position Test tallies the observed distribution of values for each position within the draw. Each of these distributions is then compared with the expected.

